EXHIBIT 3
Part 1 of 4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ARISTA NETWORKS, INC.
Petitioner

v.

CISCO TECHNOLOGY, INC.
Patent Owner

Case No.: IPR2016-00244
Patent 7,953,886

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,953,886

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Submitted by:

Keker & Van Nest LLP
633 Battery Street
San Francisco, CA 94111
(415) 391-5400 (tel)
(415) 397-7188 (fax)
Counsel for Petitioner Arista Networks, Inc.

November 24, 2015

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Federal Cases

Federal Statutes

Federal Regulations

EXHIBIT LIST

Ex. No.	Description
1001	U.S. Patent No. 7,953,886 to Bansal et al., issued May 31, 2011 ("'886 patent").
1002	U.S. Patent No. 7,200,548 to Courtney, issued April 3, 2007 ("Courtney").
1003	U.S. Patent No. 8,296,400 to Gorthy et al., issued October 23, 2012 ("Gorthy").
1004	U.S. Patent No. 7,155,496 to Froyd et al., issued December 26, 2006 ("Froyd").
1005	JUNOScript API Guide Release 5.1 (2d ed.), Nov. 6, 2001 ("JUNOScript Guide").
1006	Honeywell Bull, Multics: Commands and Active Functions (Feb. 1985).
1007	Dennis M. Ritchie & Ken Thompson, The UNIX Time-Sharing System, 17 Communications of the ACM 365 (Jul. 1974).
1008	OpenVMS User's Manual, Version 7.1 (Nov. 1996), http://www.mi.infn.it/~calcolo/OpenVMS/ssb71/6489/6489p.htm.
1009	Brian W. Kernighan & Rob Pike, The UNIX Programming Environment (1984).
1010	Byung-Joon Lee, et al., *X-CLI: CLI-Based Management Architecture Using XML* (2003) ("Lee").
1011	Alfred V. Aho et al., The AWK Programming Language (1988).
1012	Cisco's Preliminary Claim Construction Disclosure, *Cisco Systems, Inc. v. Arista Networks, Inc.*, No. 5:14-cv-05344-BLF (N.D. Cal. Aug. 24, 2015).
1013	McGraw-Hill Dictionary of Scientific and Technical Terms, 6th ed. (2003).
1014	Erik T. Ray, *Learning XML*, 2d ed. (O'Reilly Media, Inc. 2003).
1015	JUNOScript API Guide and Reference Release 4.3 (5d ed.), Nov. 13, 2001.
1016	James Boney, *Cisco IOS In a Nutshell*, 2d ed. (O'Reilly Media, Inc.

Ex. No.	Description
	2005).
1017	Dictionary of Computer Science, Engineering, and Technology (2010).
1018	Declaration of Douglas W. Clark, Ph.D.

I. INTRODUCTION

Arista Networks Inc. ("Petitioner") petitions for *inter partes* review of claims 1-10 ("the challenged claims") of United States Patent No. 7,953,886 to Bansal *et al.,* entitled "Method and System of Receiving and Translating CLI Command Data Within a Routing System" ("'886 patent") (Ex. 1001).

The '886 patent is directed to improving the user interface for Cisco routers. Historically, network engineers configured and accessed Cisco routers through the command line interface (CLI) used with Cisco's "internetwork operating system" (known as "IOS"). *Id.* at 1:12-16. As the Background section of the '886 patent explains, however, "IOS CLI is not the most program-friendly of interfaces . . . ." *Id.* at 1:26-27. By 2005, "[t]wenty years of consistency and backwards-compatibility" had resulted in a "complicated input and output scheme" that users had to "sort through . . . to input information and extract important data," and these tasks had "proven to be . . . very difficult and cumbersome . . . to automate." *Id.* at 1:26-34.

To address this limitation, the '886 patent proposes a "more structured approach to accessing and configuring a router, while still making use of the significant advantages and experience associated with IOS CLI." *Id.* at 1:34-37. This approach entails receiving commands configured in an XML format, rather than the cumbersome CLI format, and using a CLI parser to translate the XML commands

into CLI commands that the router can process. Similarly, when an operation invoked by a CLI command generates an output message (*i.e.*, text received in response to sending a command to the router), the output message is translated from a CLI format to an XML format and transmitted to a remote computer, such as the computer from which the XML-formatted command was received. *Id.* at 7:59-61.

The '886 patent was not inventive when it was applied for in 2005. By January 2001—over four years earlier—Cisco's competitor Juniper Networks, Inc. had released an XML-based interface for use with its routers, allowing administrators to access and configure the routers using XML commands instead of CLI. And in the same year, network-software company Intelliden filed patent applications praising the "user-friendly XML-based interface" of the Juniper Networks routers, while noting that "Cisco™ routers are notoriously difficult to configure" due to their "cumbersome command line interface (CLI)." Ex. 1002 at 1:47-52; *see also* Ex. 1003 at 1:47-51. Intelliden's applications disclosed a means for converting XML-based commands to CLI commands, and converting CLI output to XML, for use with Cisco routers, as Cisco claimed years later in the '886 patent. And others were also developing XML interfaces for existing CLI systems before Cisco applied for the '886 patent.

For these reasons, the Board should institute IPR of all the challenged claims, and should find them invalid on the grounds set forth below.

II. MANDATORY NOTICES (37 C.F.R. § 42.8(a)(1))

Pursuant to 37 C.F.R. § 42.8, Petitioner provides the following mandatory disclosures:

A. Real Parties-In-Interest (37 C.F.R. § 42.8(b)(1))

The real party-in-interest is Petitioner ARISTA NETWORKS, INC.

B. Related Matters (37 C.F.R. § 42.8(b)(2))

The '886 patent is at issue in *Cisco Systems, Inc. v. Arista Networks, Inc.*, No. 5:14-cv-05344 (N.D. Cal).

C. Designation of Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3))

Petitioner appoints Leo Lam (Reg. No. 38,528) of Keker & Van Nest LLP as lead counsel, and appoints Eugene M. Paige (Reg. No. 55,519), Robert A. Van Nest (*pro hac vice* motion to be filed), Brian L. Ferrall (*pro hac vice* motion to be filed), and David J. Silbert (*pro hac vice* motion to be filed) of Keker & Van Nest LLP as back-up counsel. An appropriate Power of Attorney is filed concurrently herewith.

D. Service Information (37 C.F.R. § 42.8(b)(4))

Service of any documents to lead or back-up counsel may be made to:

Keker & Van Nest LLP
633 Battery Street
San Francisco, CA 94111
(415) 391-5400 (telephone); (415) 397-7188 (fax)

Petitioner consents to service by email at the following addresses: aris-taipr@kvn.com; llam@kvn.com; dsilbert@kvn.com.

III. STANDING (37 C.F.R. § 42.104(a))

Petitioner certifies that (1) the '886 patent is available for *inter partes* review; and (2) Petitioner is not barred or estopped from requesting *inter partes* review of any claim of the '886 patent on the grounds identified herein. This petition is filed in accordance with 37 C.F.R. § 42.106(a). Concurrently filed herewith is an Exhibit List per 37 C.F.R. § 42.63(e). The Office is authorized to charge the required fee as set forth in 37 C.F.R. § 42.15(a) to Deposit Acct. No. 506260. The Office is further authorized to charge fee deficiencies and credit overpayments to the above-referenced Deposit Account.

IV. OVERVIEW

A. The alleged invention of the '886 patent

The '886 patent (Ex. 1001), entitled "Method and System of Receiving and Translating CLI Command Data Within a Routing System," was filed on July 8, 2005 and issued on May 31, 2011. The patent is directed to improving the user interface for Cisco routers. Historically, network engineers configured and accessed Cisco routers through the command line interface (CLI) used with Cisco's "internetwork operating system" (known as "IOS"). *Id.* at 1:12-16. As the Background section of the '886 patent explains, however, "IOS CLI is not the most program-friendly of interfaces . . . ." *Id.* at 1:26-27. By 2005, "[t]wenty years of consistency

and backwards-compatibility" had resulted in a "complicated input and output scheme" that users had to "sort through . . . to input information and extract important data," and these tasks had "proven to be . . . very difficult and cumbersome . . . to automate." *Id.* at 1:26-34.

To address this limitation, the '886 patent proposes a "more structured approach to accessing and configuring a router, while still making use of the significant advantages and experience associated with IOS CLI." *Id.* at 1:34-37. This approach entails receiving commands configured in an XML format, rather than the cumbersome CLI format, and using a CLI parser to translate the XML commands into CLI commands that can be processed by the router. Table 1 of the '886 patent provides an example of such an input command in XML format:

TABLE 1

```
<add>
<k_mpls_label>
<k_range>
<range-min>10</range-min>
    <range-max>300</range-max>
<k_static>
    <static-min>30</static-min>
    <static-max>150</static-max>
</k_static>
</k_range>
</k_mpls_label>
</add>
```

Id. at Table 1.

The CLI command corresponding to this example is "mpls label range 10 300 static 30 150." *Id.* at 5:57-65. The XML tags shown in Table 1—which are de-

limited by angle brackets ("<" and ">")—contain "CLI keywords" or enclose "command parameters," both of which the parser extracts and arranges into CLI commands. *Id.* at 5:5-6 and 5:30-36; Declaration of Douglas W. Clark, Ph.D. ("Clark Decl.") (Ex. 1018) ¶ 19. In this example, "mpls label," "range," and "static" are CLI keywords, each contained in the tags with the corresponding text; "10" and "300" are command parameters enclosed respectively by the tags "range-min" and "range-max"; and "30" and "150" are command parameters enclosed respectively by the tags "static-min" and "static-max." Clark Decl. ¶ 19.

The '886 patent further describes that, when an operation invoked by a CLI command generates an output message (*i.e.*, text received in response to sending a command to the router), the output message is translated from a "CLI format into an XML format having the CLI syntax." Ex. 1001 at 7:59-61. To perform this translation, the system parses the output message "to identify at least one CLI token," then "translat[es] each CLI token of the output message into a corresponding XML value according to a stored mapping of CLI tokens-to-XML values." *Id.* at 7:62-66. The translated output message, now in XML format, is then transmitted to a remote device external from the routing system, such as the computer that sent the input command. *Id.* at 8:1-3.

B. The State of the Art

1. Command Line Interface (CLI)

A command line interface (often abbreviated "CLI") is a type of user interface to a computer system in which the user types commands in response to a prompt and receives responses from the system on subsequent lines. Clark Decl. ¶ 21. Command line interfaces have long been well-known in the art. *Id.* For example, the influential MULTICS system, developed at MIT in the 1960's, used a CLI, as did Unix, developed in the 1970's, and Digital Equipment Corporation's Digital Command Language or DCL, developed in the 1980's or earlier. *Id.*; Honeywell Bull, Multics: Commands and Active Functions (Feb. 1985) (Ex. 1006) at 21 (MULTICS); Dennis M. Ritchie & Ken Thompson, The UNIX Time-Sharing System, 17 Communications of the ACM 365 (Jul. 1974) (Ex. 1007) at 7; OpenVMS User's Manual, Version 7.1 (Nov. 1996) (Ex. 1008) at 17, Section 3.2, 18, Section 3.3 and 19, Section 3.3.3 (DCL).

CLIs are text-based interfaces and generally operate in a similar manner. Clark Decl. ¶ 22. A user is presented with a command prompt (*e.g.*, ">" or "C:\>"), after which the user may type a command requesting that the system perform a particular operation. *Id.* The user must generally know what commands are valid for the system in question, as well as the required syntax for parameters associated with those commands. *Id.* For example, in Unix, a user can type the command "ls -

l" to cause the operating system to show a long-form list of the contents of the current directory. *Id.*; Brian W. Kernighan & Rob Pike, The UNIX Programming Environment (1984) (Ex. 1009) at 19.

In response to the command, the computer system will typically return some text that constitutes the response to the command, or some other text to indicate that an error occurred (perhaps, for example, a misspelling or mis-formatting of the command). Clark Decl. ¶ 22. The user is then returned to the command prompt, where another command can then be entered. *Id.*

Cisco's Internetwork Operating System (IOS) had been in use for over twenty years when the '886 patent was applied for. *See* Ex. 1001 at 1:27-30. It uses a CLI that is similar to ones used in the prior art. Clark Decl. ¶ 23. IOS software runs on Cisco routers, which are devices that enable packetized information to be passed from one machine, through one or more routers, to a destination machine. James Boney, *Cisco IOS In a Nutshell*, 2d ed. (O'Reilly Media, Inc. 2005) (Ex. 1016) at 9. The routers are accessed and configured by administrators via a serial port, ssh, or other connection from a remote computer. *See id.* at 16-17, 32. Using the CLI, the administrator can access and configure the router by entering textual commands that follow the proper syntax. *See id.* at 9. Just as with prior art CLIs, in response to an entered command, the router may respond with requested infor-

mation, make some change in the router's settings, or, in the event of an incorrect command, report an error.

2. Extensible Markup Language (XML)

Extensible markup language, or "XML," is an open standard markup language that was developed in the 1990s. Erik T. Ray, *Learning XML*, 2d ed. (O'Reilly Media, Inc. 2003) (Ex. 1014) at 16. A "markup language" is a language that allows users to annotate digital documents, usually by embedding the annotations in the document's text. Clark Decl. ¶ 24. These annotations can be used for many purposes. For example, they may instruct a web browser to display certain text in a particular way, or to make a particular string of text a hypertext link. *Id.* Annotations may also describe portions of a document—for example, they may label certain text as a street name within an address, or as a citation, and software may then use this information in a variety of ways. Ex. 1014 at 24-26; Clark Decl. ¶ 24.

Like other markup languages, XML features "tags" embedded within a document. Ex. 1014 at 57. Tags are delimited by angle brackets ("<" and ">"). *Id.* at 11. With certain exceptions, tags include a start tag and an end tag (in which the name of the tag is preceded by a slash). An XML "element" consists of a start tag, a corresponding end tag, and everything in between (except in the case of an "emp-

ty element," which is discussed below). Clark Decl. ¶ 25; *see also* Ex. 1014 at 58. For example, an XML element might consist of:

<street address>2800 S. Randolph Street</street address>

XML tags may also contain "attributes" that describe the element. Clark Decl. ¶ 25; *see also* Ex. 1014 at 67. For example, an XML element might consist of:

<name language="french">Pierre-Auguste</name>

In this example, "language" is the name of the attribute and "french" is the attribute's value. Elements may also be nested within other elements. Clark Decl. ¶ 25; *see also* Ex. 1014 at 58. For example, an XML element might consist of:

```
<name language="french">
        <first name>Pierre-Auguste</first name>
        <last name>Renoir</last name>
</name>
```

In this example, the "first name" and "last name" elements are nested within the "name" element. Clark Decl. ¶ 25. As noted above, XML also allows "empty elements," which are elements with start tags only, containing no nested elements or other contents, and whose start tags are terminated by a slash character. *Id.* Empty elements have attributes within their start tags. Ex. 1014 at 66. For example, an XML element may consist of:

<artist name="Pierre-Auguste Renoir"/>

Clark Decl. ¶ 25.